N 71 11125

NASA TECHNICAL MEMORANDUM

NASA TM X- 64527

1970

NASA TM X- 64527

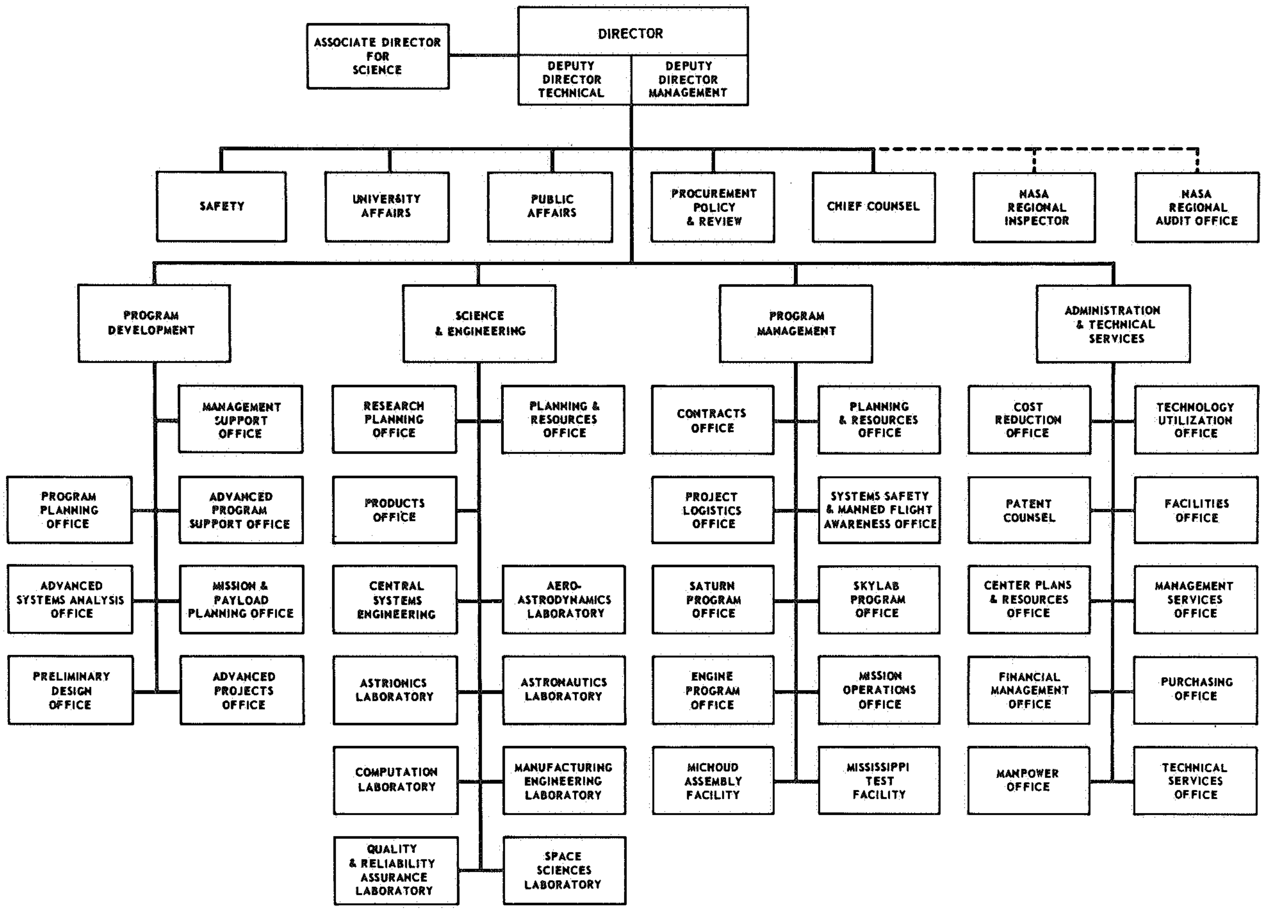
COMPUTATION RESEARCH AT MSFC

RESEARCH ACHIEVEMENTS REVIEW

VOLUME III REPORT NO. 12

SCIENCE AND ENGINEERING DIRECTORATE
GEORGE C. MARSHALL SPACE FLIGHT CENTER
MARSHALL SPACE FLIGHT CENTER, ALABAMA

GEORGE C. MARSHALL SPACE FLIGHT CENTER



RESEARCH ACHIEVEMENTS REVIEWS COVER
THE FOLLOWING
FIELDS OF RESEARCH

- Radiation Physics
- Thermophysics
- Chemical Propulsion
- Cryogenic Technology
- Electronics
- Control Systems
- Materials
- Manufacturing
- Ground Testing
- Quality Assurance and Checkout
- Terrestrial and Space Environment
- Aerodynamics
- Instrumentation
- Power Systems
- Guidance Concepts
- Astrodynamics
- Advanced Tracking Systems
- Communication Systems
- Structures
- Mathematics and Computation
- Advanced Propulsion
- Lunar and Meteoroid Physics

NASA TM X-64527

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D. C.

RESEARCH ACHIEVEMENTS REVIEW
VOLUME III REPORT NO.12

COMPUTATION RESEARCH AT MSFC

SCIENCE AND ENGINEERING DIRECTORATE
GEORGE C. MARSHALL SPACE FLIGHT CENTER
MARSHALL SPACE FLIGHT CENTER, ALABAMA

1970

UNITS OF MEASURE

In a prepared statement presented on August 5, 1965, to the U. S. House of Representatives Science and Astronautics Committee (chaired by George P. Miller of California), the position of the National Aeronautics and Space Administration on Units of Measure was stated by Dr. Alfred J. Eggers, Deputy Associate Administrator, Office of Advanced Research and Technology:

"In January of this year NASA directed that the international system of units should be considered the preferred system of units, and should be employed by the research centers as the primary system in all reports and publications of a technical nature, except where such use would reduce the usefulness of the report to the primary recipients. During the conversion period the use of customary units in parentheses following the SI units is permissible, but the parenthetical usage of conventional units will be discontinued as soon as it is judged that the normal users of the reports would not be particularly inconvenienced by the exclusive use of SI units."

The International System of Units (SI Units) has been adopted by the U. S. National Bureau of Standards (see NBS Technical News Bulletin, Vol. 48, No. 4, April 1964).

The International System of Units is defined in NASA SP-7012, "The International System of Units, Physical Constants, and Conversion Factors," which is available from the U. S. Government Printing Office, Washington, D. C. 20402.

SI Units are used preferentially in this series of research reports in accordance with NASA policy and following the practice of the National Bureau of Standards.

PREFACE

In February, 1965, Dr. Ernst Stuhlinger, now Marshall Space Flight Center's Associate Director for Science, initiated a series of Research Achievements Reviews which set forth those achievements accomplished by the laboratories of the Marshall Space Flight Center. Each review covered one or two fields of research in a form readily usable by specialists, systems engineers and program managers. The review of February 24, 1966, completed this series. Each review was documented in the "Research Achievements Review Series."

In March, 1966, a second series of Research Achievements Reviews was initiated. This second series emphasized research areas of greatest concentration of effort, of most rapid progress, or of most pertinent interest and was published as "Research Achievements Review Reports, Volume II." Volume II covered the reviews from March, 1966, through February, 1968.

This third series of Research Achievements Reviews was begun in March, 1968, and continues the concept introduced in the second series. Reviews of the third series are designated Volume III and will span the period from March, 1968, through March, 1970.

The papers in this report were presented March 26, 1970

William G. Johnson
Director
Research Planning Office

SIMULATION, MATHEMATICS, AND LANGUAGE

By G. Reisig

OPTICAL PROBE FOR VISUAL SIMULATION

LIST OF TABLES

LIST OF ILLUSTRATIONS

CONTENTS (Continued). . .

COMPUTER-GENERATED VIEW OF A LUNAR LANDSCAPE

LIST OF ILLUSTRATIONS

HYBRID SIMULATION OF VEHICLE RESPONSE TO MANY MEASURED WINDS

TELEMETRY LANGUAGE SYSTEM

LIST OF ILLUSTRATIONS

SOME EXPERIMENTAL RESULTS CONCERNING THE ERROR PROPAGATION IN RUNGE-KUTTA TYPE FORMULAS

MARSHALL VEHICLE ENGINEERING SIMULATION SYSTEM (MARVES)

By R. N. Setter

LIST OF TABLES

LIST OF ILLUSTRATIONS

SIMULATION, MATHEMATICS, AND LANGUAGE

By

G. Reisig

The utility of any computer model or computer system can be measured only to a limited degree by its computational speed and task capacity. After a stormy and virulent period of approximately two decades of hardware development, the value and the merits of any computer concept are assessed at the present state of development more and more in terms of applicability to particular classes of mathematical and physical problems and in terms of the efficiency in solving these problems. It is common knowledge that software development presently lags far behind the state-of-the-art of computer hardware. Prominent categories of software techniques are numerical mathematics and computer languages. Numerical methods can be and must be advanced according to the algorithmic capabilities of present computer design concepts. Problem languages are particularly efficient means of increasing the volume of solvable mathematical and physical problems and of enlarging the parameter variability by several orders of magnitude.

Another complex of computer utilization within the present status of sophisticated hardware is the field of simulation, both with analog and digital techniques and with the symbiosis of both, the hybrid techniques. The available computational speeds permit on-line simulation of very complex mathematical and physical models, such as whole rocket systems, and even the simulation of computer processes themselves.

The papers of this Research Achievements Review cover six topics that pertain to the quoted field of computer simulation, mathematics, and language. Brief abstracts for the six papers are included in this introduction prior to the presentation of the papers themselves.

OPTICAL PROBE FOR VISUAL SIMULATION

The training of astronauts for controlling or driving vehicles, either landing craft or surface vehicles, necessitates simulation devices to provide realistic environmental conditions. For the man-vehicle relations involved in lunar rover driving, a model of the lunar surface is sensed by a television camera, which is the optical probe for visual simulation. The television circuit is completed to a closed loop by the driver's vision of the lunar surface picture. He, in turn, activates the controls for orienting the television camera onto the intended driving course of the lunar surface model by using the rover driving controls. The immediate problem with this simulation device concerns the adequate size of the field-of-view "out of the window", as presented on the television screen. A simultaneous problem is the limitation of the field-of-depth, resulting from the scaled-down terrain model. The pursued approach to solving the complex problem of expanding the field-of-depth utilizes the Scheimpflug condition, which optimizes the mutual orientation of the planes of the object, the optical system, and the image.

COMPUTER GENERATED VIEW OF A LUNAR LANDSCAPE

The utility of closed circuit television systems for simulating a driving environment is limited, since the utility depends on the given physical model of the environment. A much increased variability of the environment and flexibility of the out-of-the window display for the driver can be accomplished if the environmental images are produced by a computer. A large variety of lunar surface features, including rocks and craters, can be analytically generated. Beyond that, a wealth of driving flexibility can be provided by analytically modifying the influence of aspect, of occultation, of sun elevation, and of vehicle shadow.

This digitally controlled display of the lunar landscape operates on real time and is particularly suited to training on the lunar roving vehicle simulator.

HYBRID SIMULATION OF VEHICLE RESPONSE TO MANY MEASURED WINDS

The objective of this task is to perform a statistical analysis of vehicle response to very large sets of actual wind data. This work has been performed on a special purpose high speed analog computer, since digital costs are prohibitive. Because of equipment limitations, it has been necessary to simplify the model and run the same winds again and again, and then compile the statistics largely by hand. The next phase of the project is concerned with the expansion of the model to a full six-degree-of-freedom, nonrigid body with highly nonlinear aerodynamics, complete automation of the statistical analysis, and a short turnaround time for a set of 2000 measured winds, so that parameter studies may be performed. It appears logical and feasible to perform these computations on a modern hybrid computer. With the experience gained, a run can now be completed within 3 hours. The analog portion is capable of much higher speeds, and the knowledge is now available to speed up the digital portion by at least a factor of two whenever the mathematical model has become sufficiently stabilized to allow detailed machine-language programming. Many jobs now considered impossible for reasons of time or cost will become quite practical, and the computation customer may search for new applications.

TELEMETRY LANGUAGE SYSTEM

A Telemetry Language System is being developed for the Instrumentation Checkout Complex of the Quality and Reliability Assurance Laboratory of Marshall Space Flight Center. This system will provide a language to be utilized by both telemetry engineers and programmers in writing telemetry checkout procedures. Even though the language is the key feature, the software system will provide a complete range of services including executive, loader, compiler, and all utility and support routines.

The Telemetry Language System will be capable of acquiring and recording any signal that can be received and routed by the Automatic Telemetry Ground Station. Evaluation of the telemetry data will be limited initially to post-test data.

The telemetry language is substantiated by means of some forty "operators". The desired telemetry checkout program is composed of these operators. The savings in man-time with this procedure in telemetry language as compared to programming in assembly language amounted to a ratio of 1 to 3000 in a sample case.

SOME EXPERIMENTAL RESULTS CONCERNING THE ERROR PROPAGATION IN RUNGE-KUTTA TYPE FORMULAS

The Runge-Kutta method is a particularly efficient, flexible, and extensively used tool for the numerical integration of differential equations. However, for optimizing the necessary computational effort, an adequate control of the integration procedure must be provided in terms of error-propagation analysis.

Two approaches for the global error-propagation in Runge-Kutta type formulas are presented in this review. The first represents an essentially conventional approach in which the matrix of the partial derivatives of the differential equations and the local truncation and roundoff errors must be available. Approximate values for these local errors are discussed.

The second approach is based on two-sided Runge-Kutta formulas. The method does not require any partial derivatives of the differential equations. However, in this case, two integration runs are necessary for each point of the function to be integrated.

If properly operated, both approaches will lead to upper and lower bounds for the solution of the integration problems. Two illustrative examples are presented for both approaches.

MARSHALL VEHICLE ENGINEERING SIMULATION SYSTEM (MARVES)

One of the basic integral elements of system analysis of rocket and space vehicles is the establishment of theoretical flight trajectories. Throughout the history of rocket and space vehicle project development, the methods of mathematically designing a trajectory have reached a status of standardization. Thus, the trajectory designer now has available a set of mathematical standard procedures and other sets of physical parameters characterizing the vehicle system and the external flight conditions. The purpose of MARVES is to provide a software system or trajectory building language that enables the trajectory designer to select the most suitable elements of the mathematical procedures and physical parameters to establish an optical trajectory for a specified vehicle mission. The available mathematical procedures comprise typical differential equations and types of integration techniques. The physical parameter sets provide the flight environment, the vehicle structure including vehicle aerodynamics, and vehicle propulsion including guidance and control concepts. Thus, a large variety of combinations of these elements is available to the trajectory designer in the mode of man-computer conversation. The problem-oriented conversational language is presently being developed, and although the subroutine library for physical trajectory parameters was originally developed for Saturn trajectory simulation, it is now being extended to meet Space Shuttle and Space Station requirements.

OPTICAL PROBE FOR VISUAL SIMULATION

By

W. Polstorff

Work in the Simulation Branch of the Computation Laboratory of Marshall Space Flight Center is applied primarily to the modeling of control systems. The system's blocks are programmed on a hybrid computer and interconnected in much the same way as will be done in the real system. The validity of concepts is checked, the performance is optimized, details are added, the influences of parameter tolerances and of failure modes are investigated, and on-the-spot modifications are introduced. The independent variable in such a system is time. Time can be scaled as desired, and quite often real time is chosen. In problems that require a large number of solutions, such as in statistical evaluations, high running speed is important resulting in time scales of 10:1 and faster.

Man-vehicle simulations represent a large portion of the work load in the Simulation Branch. The use of real time is essential in these simulations. It can be said, in computer terms, that man-vehicle simulation problems differ from other problems in their input/output devices. Of these devices the view "out of the window" is particularly demanding.

Model closed-circuit television systems are most often used for visual simulation. Such a system is used in the Lunar Rover program where the requirements for visual cues in lunar driving are as follow:

1. A wide field-of-view including a look to the side of the vehicle as well as a front view for navigating around obstacles.

2. A proper view to the rear, to allow controlled backing if required.

3. Adequate resolution with a maximum field-of-depth enabling identification of distant landmarks as well as the judging of terrain properties near the rover from their appearance.

4. Similar hard contrast between light and shadow as experienced on the moon.

5. Interactions of the vehicle with the soil, such as dust clouds and vehicle tracks, should be visible.

Only part of these requirements can be met with existing equipment.

The Lunar Rover Vehicle simulation uses an SMK-23, a pilot trainer for landing and takeoff acquired from the Air Force and modified for the lunar driving simulation. (See Table 1.) Figure 1 shows the computer complex, the SMK-23, Link's 6-DOF, the driver at the controls, and the interconnections. The camera moves across the terrain according to the commands of the driver, which are converted by the computer into positioning commands for the camera. Four sensors attached at the camera optics provide the computer with terrain information. The computer derives the height of the driver's pupil point and the attitude (pitch and roll) of the vehicle, and generates the related command signals for the z-servo, the pitch servo, and the roll servo. Similarly, the proper command signals are generated for the motion system. Figure 2 is a schematic of the probe, which is called an articulated probe because the changes in the viewing direction are accomplished by rotating prisms and mirrors while the camera maintains its orientation perpendicular to the terrain model. The out-of-the-window view is presented by a television monitor that is equipped with a precision infinity display optics in front.

By comparing the performance of the SMK-23 as shown in Table 1 with the requirements, the following shortcomings are found:

1. Inadequate field-of-view.

2. Lack of rear view.

3. Inadequate field-of-depth.

4. Inadequate contrast.

5. The size of the terrain represented is limited.

TABLE 1. PERFORMANCE OF THE SMK-23 IN LUNAR ROVER SIMULATION

a. Servo Performance, Terrain Scale 100:1

Translation			Rotation		
Range	Velocity	Acceleration	Angle	Rate	Acceleration
X 800 m	15 km/hr	0.9 g	θ pitch ± 25°	± 64°/sec	± 100°/sec^2
Y 360 m	15 km/hr		ϕ roll ± 60°	± 172°/sec	± 500°/sec^2
Z 20 m	1.8 km/hr	0.5 g	ψ yaw control	± 115°/sec	± 500°/sec^2

b. Optical Performance

Field-of-View: 38° vertical by 50° horizontal
Field-of-Depth at f/55 ∞ ÷ 25 cm or in 100:1 Scale: ∞ ÷ 25 m
Aperture: d = 0.6 mm (diffraction limit for 0.1°, d = 0.4 mm)
Scan Lines: 1023
Television Resolution: 600 lines or 12 lines/deg

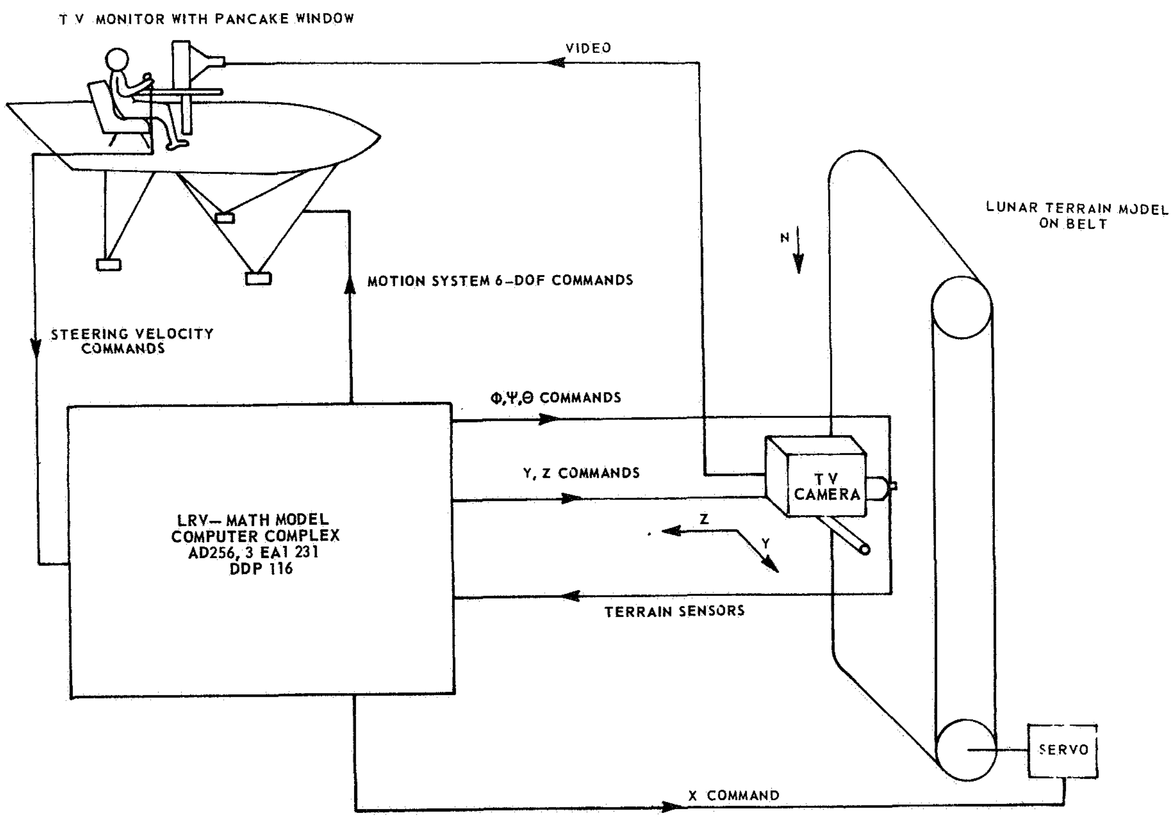


Figure 1. Sketch of the Lunar Rover simulation.

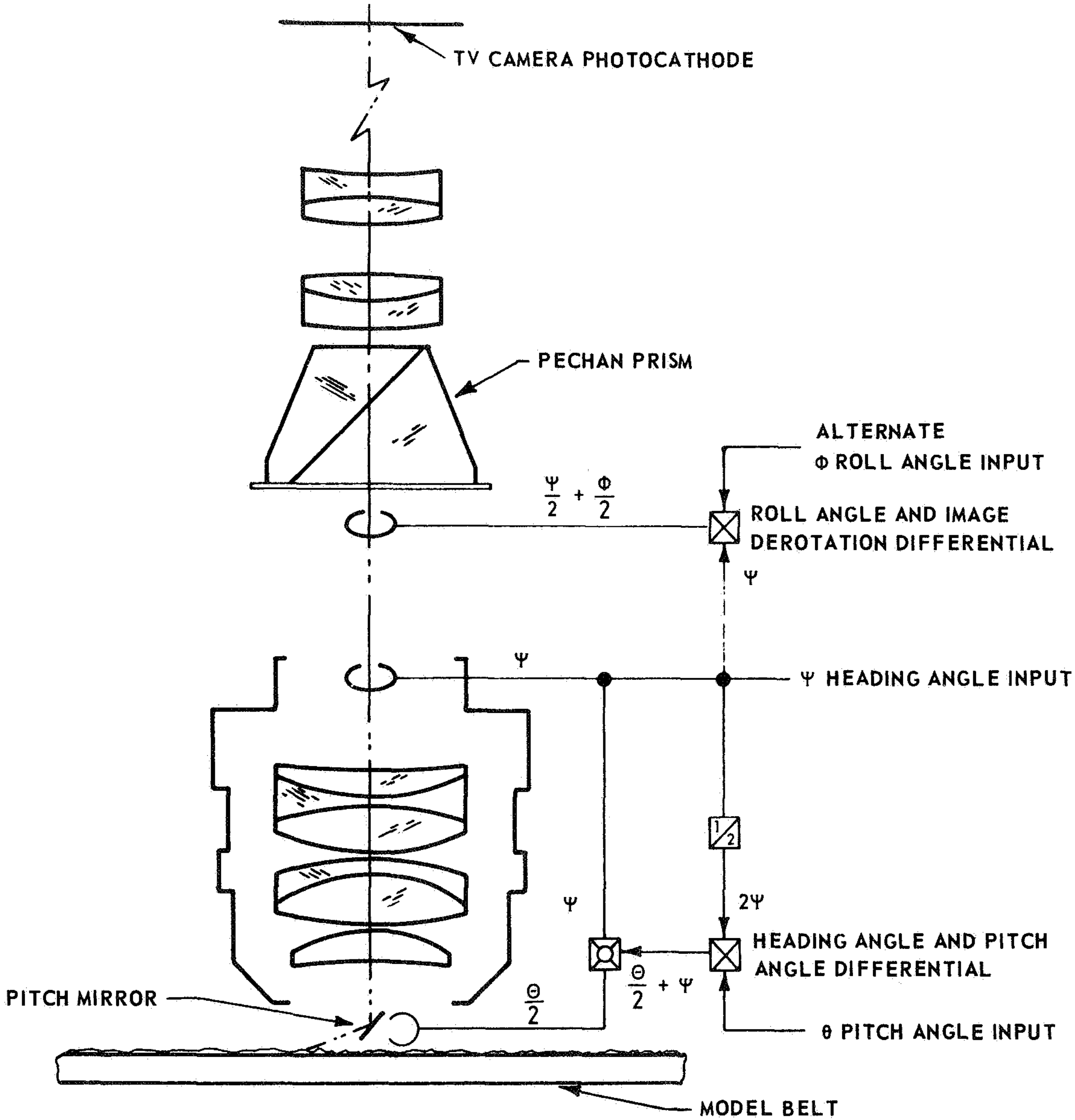


Figure 2. Typical optical probe assembly schematic.

6. No tracks and no dust clouds are generated.

The first and third deficiencies are of general interest not only to NASA but also to the Air Force, the airlines, and the aircraft industry. These deficiencies were the subject of a research contract awarded to Goodyear Aerospace Corporation by Marshall Space Flight Center. Goodyear's efforts do not represent the only attack on this problem, however; they were preceded by others who attempted to increase the field-of-view. Using the conventional optical design of articulated single-barrel probes, a field-of-view of 95 degrees in the diagonal was reached. (This compared with 60 degrees for the SMK-23.) However, because of the limited detail transmitted by a television channel, the increase in field-of-view is accomplished by a decrease in angular resolution.

Marquardt, now Conductron, promoted an ambitious concept for a wraparound display combining hyperboloidal and ellipsoidal mirrors (Fig. 3). Rotations of the entire pickup around the entrance pupil are necessary to change the viewing direction. The primary limitation of the system is the use of a single camera/projector television channel with moderate resolution to depict a full 360 degrees of scenery. Also, only an

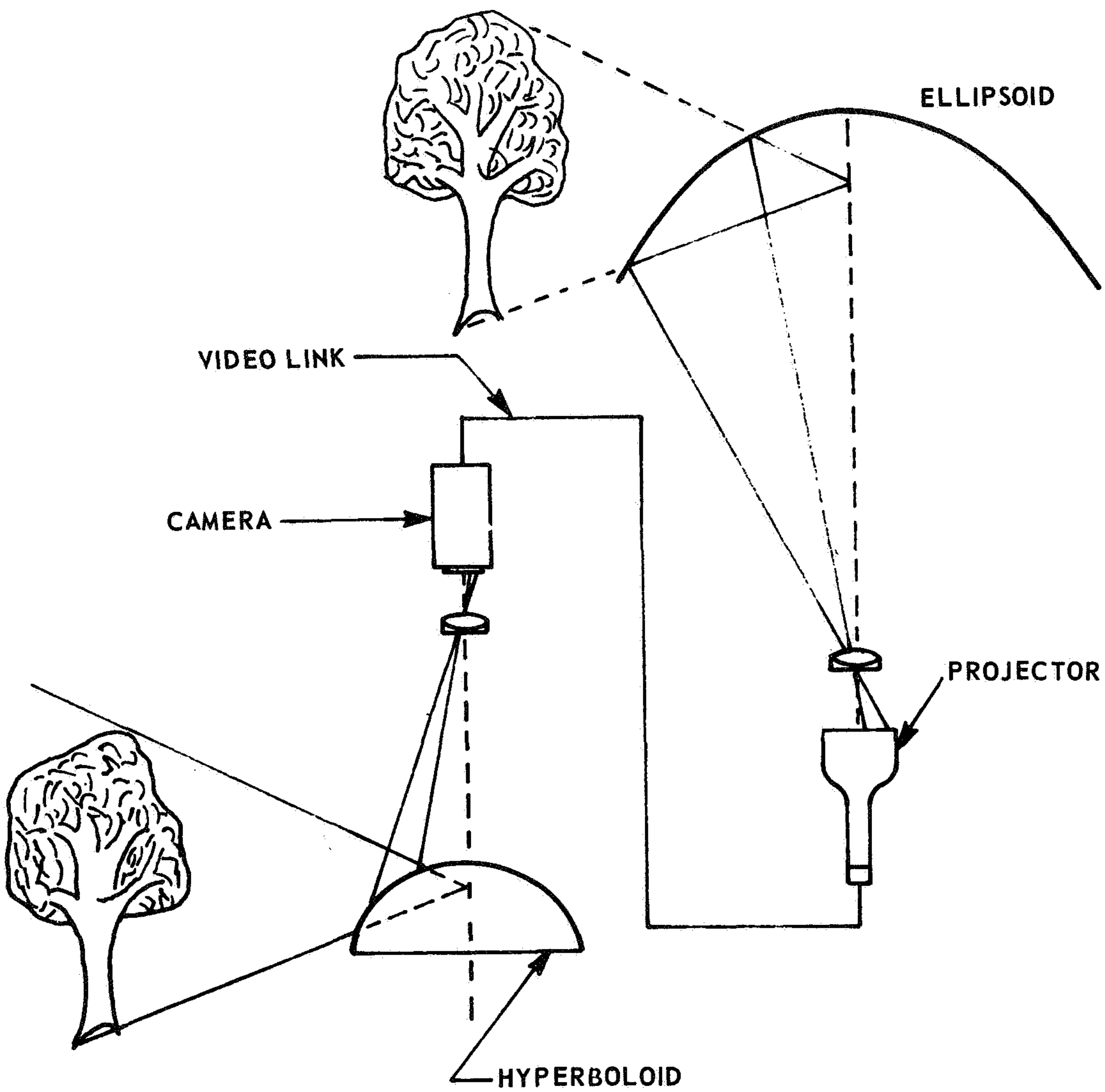


Figure 3. The VUEMARQ system.

outer ring of the camera cathode carries useful video information. An additional disadvantage is the location of the pupil point inside the front optics, which limits the closest approach of the look point to the model.

A different approach was pursued by Dalto using a camera with a fisheye front lens. Looking vertically down to the terrain model, the field-of-view is all around in the horizontal; and in the vertical, the field-of-view is from straight down to 20 degrees above the horizon. Thus, slightly more than the hemisphere is imaged on the face of the camera tube. The view required is obtained by scanning only that part of the camera cathode that contains this video information. The scan is moved electrically according to the change in viewing direction. The proper geometry is restored by raster manipulation. Rotations of optical elements are no longer required. Size and location of windows can be programmed, thus providing ultimate flexibility in visual simulation. However, the Dalto system had the following disadvantages:

1. Astigmatism, sagittal, and meridional focuses are separated resulting in poor resolution; at 90 degrees off axis, in the horizontal viewing direction, the resolution is only two television lines/degree.

2. The point of perspective depends on its location in the viewing direction, causing objects to change their shape on approach.

3. The look point is inside the front optics.

4. The electronics system is quite complex.

This system was used in Langley's Lola System (a lunar mission simulator).

Under contract to Trans-World Airlines, Dalto has developed and built another wide-angle visual simulator and designed a sophisticated articulated probe. The probe provides a 230-degree wide field-of-view using four television channels (Fig. 4). The front optical element of the probe is a sphere that approximates a paraboloid. The focal point is the pupil point of the system. All rays in the direction of the pupil point are reflected parallel to the optical axis into a system of mirrors and prisms, which are arranged on axes one and two. By rotating the front element with these mirrors and prisms, the viewing direction can be changed without rotating the cameras. Using four beam splitters, images of the front sphere are formed on four segments of hollow spheres. Thus, a reproduction of the view from the pupil point is generated at the focal points of each segment, where it is used to form an image of that particular section of the total view on the face of a camera cathode. The precise imaging of the front element

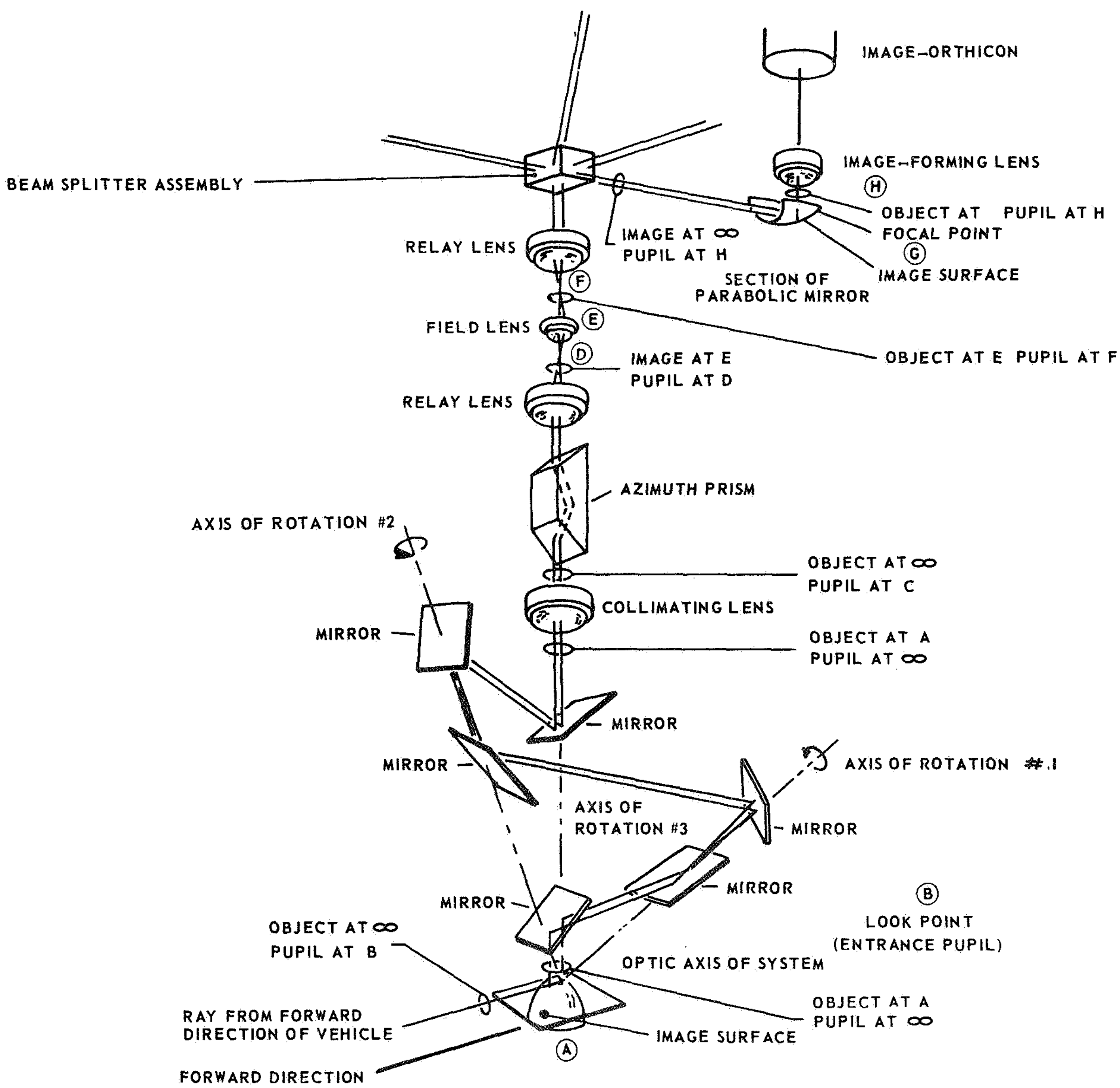


Figure 4. Wide angle pickup probe.

on the segments of hollow spheres in a dynamic situation is quite demanding. Other problems result because (1) the beam splitters cause a low light level at the cameras, (2) the axes of rotation are not perpendicular to each other, and (3) the areas of the lenses that are most distant from the axes in the imaging process are used. This system was recently delivered to TWA.

Goodyear Aerospace Corporation has used a more conventional approach to develop a probe with a wide field-of-view (Figs. 5 and 6). Three identical optical barrels are arranged above a three-faceted mirror such that the individual fields-of-view add up to a composite view of 178.5 degrees in the horizontal. In the vertical, the field-of-view measures 45.5 degrees. Because of the use of three channels, it is no longer possible to perform the angular motions of the view merely by rotating a few mirrors or prisms. Instead, it is necessary to rotate the entire assembly around the look point. However, by using this approach, the quality of the video generated by the probe will not be compromised by optical shortcomings.

The other problem that was the subject of the Goodyear study for MSFC is the improvement of the field-of-depth. The approach normally used to

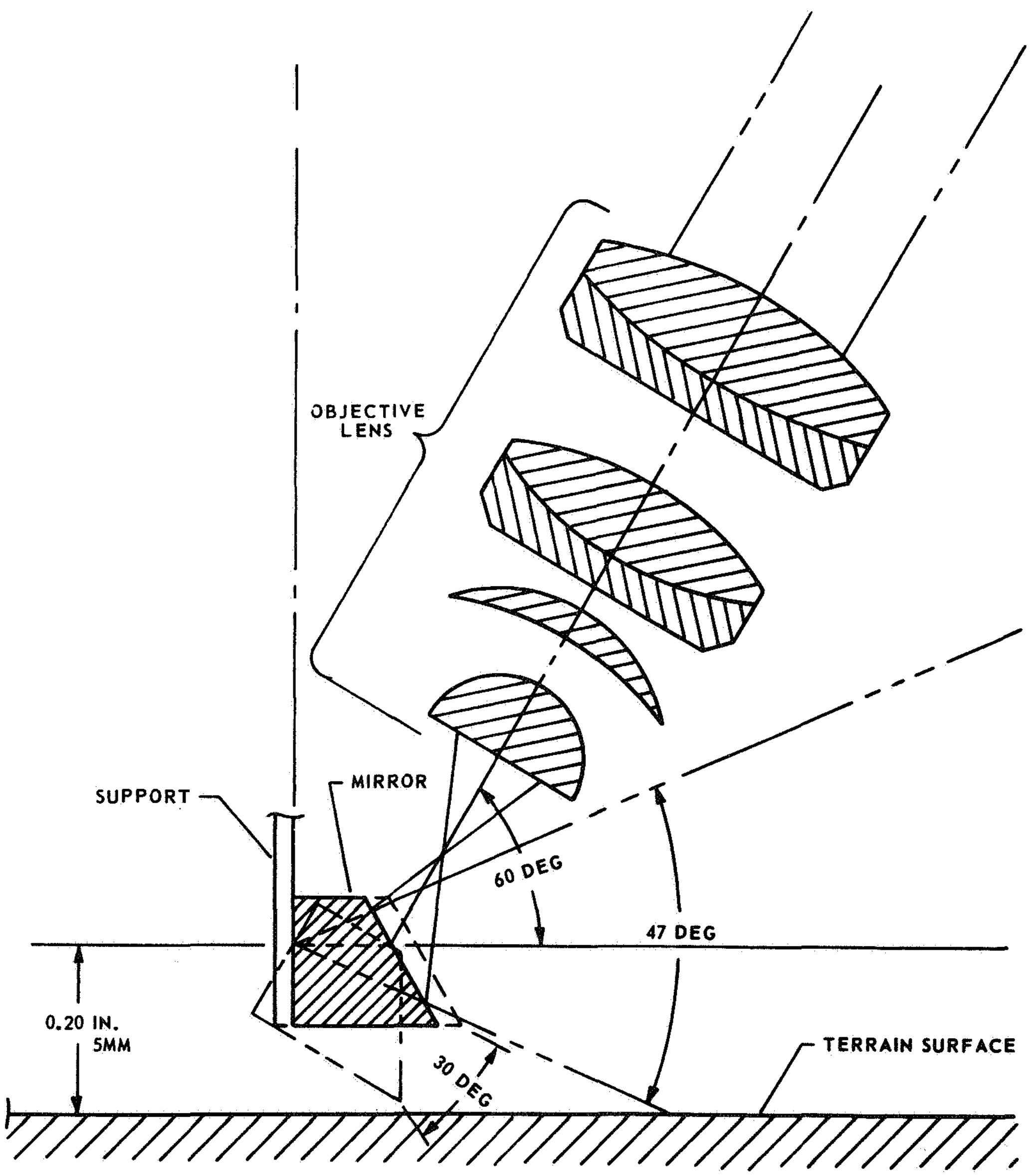


Figure 5. Objective lens mirror assembly profile (one channel).

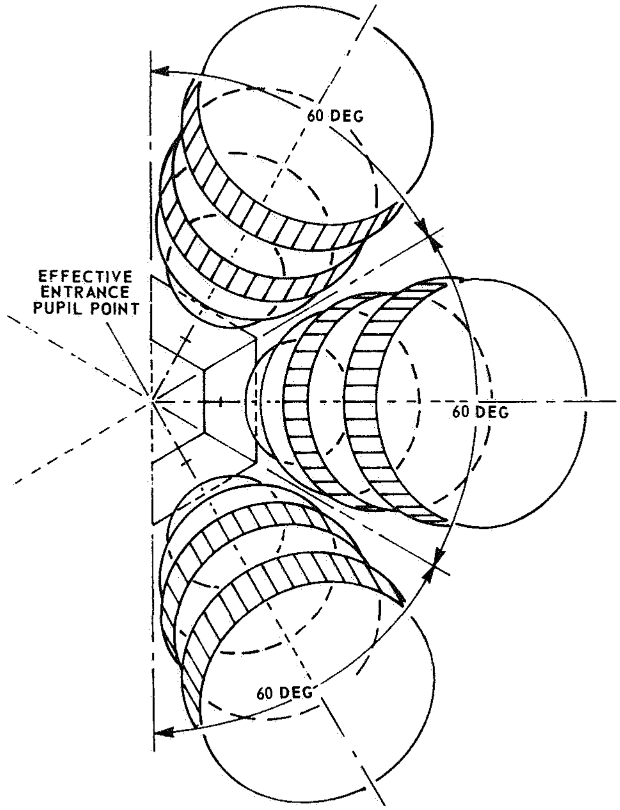


Figure 6. Objective lens mirror assembly, plan view.

increase the field-of-depth is to decrease the aperture of the imaging system. However, a limit to the smallness of the aperture is set by the desired angular resolution of 1000 television lines for a 60-degree field-of-view. This limit is a consequence of the wave nature of light and is called the diffraction limit.

The desired resolution can be obtained with an aperture of 0.4 mm diameter. The closest object with the desired angular resolution imaged with this aperture and simultaneously imaged with infinity is at a distance of about 25 cm from the front lens, independent of the focal length used. However, by limiting the task to imaging a terrain model, which for these purposes can be considered as a plane, an image of this plane can be formed that is in focus on an inclined image plane (Fig. 7). According to the Scheimpflug condition, the image plane is oriented such that it intersects with the object plane and with the principal plane of the lens in one common line. However, because of the variation in distance of the image points from the lens, the magnification varies also, thus introducing a distortion. Instead of tilting the image plane, an inclined lens can be used (Fig. 8). Goodyear actually uses a relay lens and not the objective lens as the Scheimpflug corrector lens (Figs. 9 and 10). The tilt range of this lens system is ±30 degrees. It consists of seven elements.

The probe is designed to be used in an f-stop range from f/6 to f/24 with the aperture varying from 1.5 mm to 0.4 mm. The Scheimpflug correction works best at high altitudes above the terrain. At 20 mm with the optical axis parallel to the ground, the tilt angle has reached its maximum excursion. Nearer the surface, the field-of-depth correction is not adequate. However, a substantial improvement above the present image quality can be expected, but there is a substantial increase in complexity for the added field-of-depth. As Figure 10 shows, it is necessary to arrange five different motions to maintain proper focus and registration. These are:

1. Tilting of the lens.

2. Rotation of the lens system to keep the tilt axis parallel to the ground.

3. Translation of the tilt lens in the direction of the optical axis.

4. Translation of the lens in the direction of the system's optical axis to counteract image shift.

5. Refocusing of the camera tube.

The development of the probe still requires a substantial in-house effort to mount it on gimbals and provide the necessary servos to orient the probe under computer control; but the Goodyear system has a good chance to become the first high quality, wide-angle probe in this country applicable to closed-circuit television systems for simulation. However, because of the complexity of these model closed-circuit television systems, it is natural to look for other approaches also.

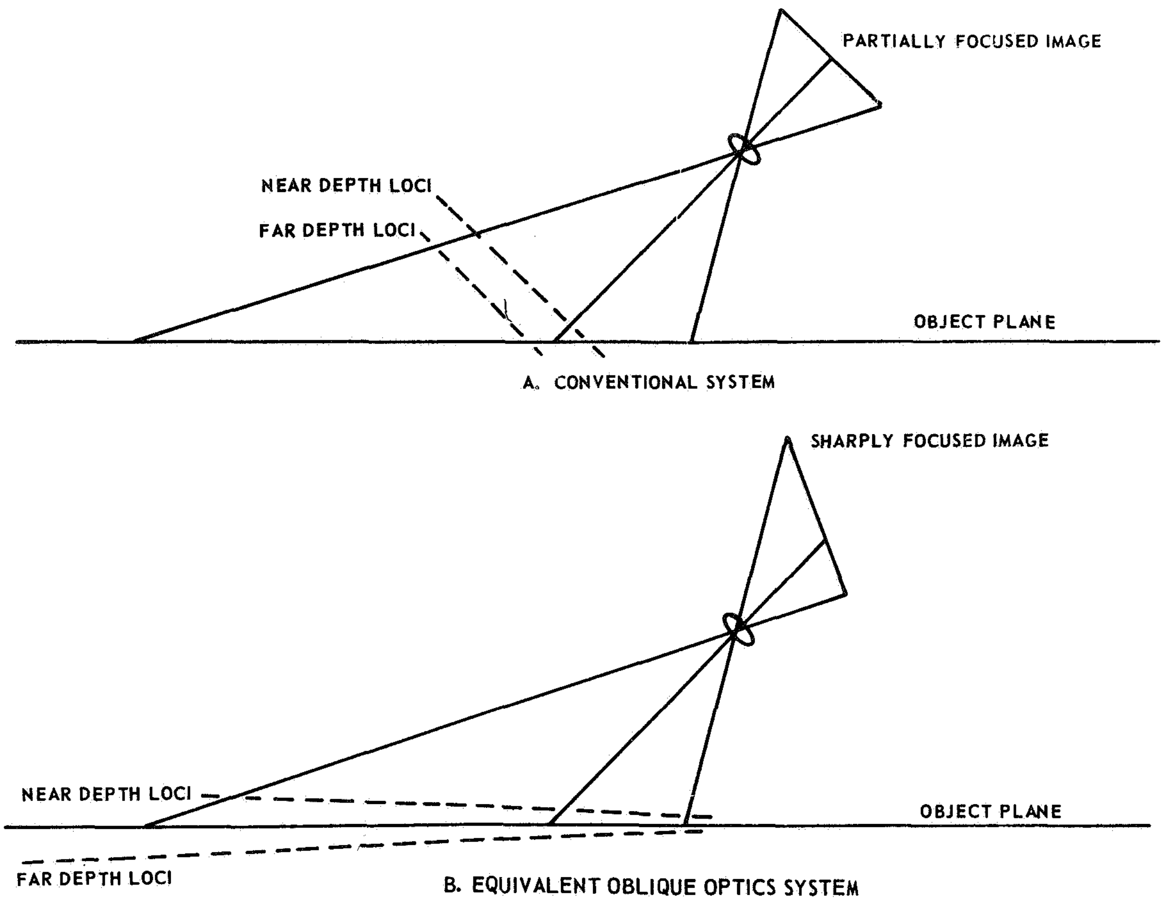


Figure 7. Comparison of image systems.

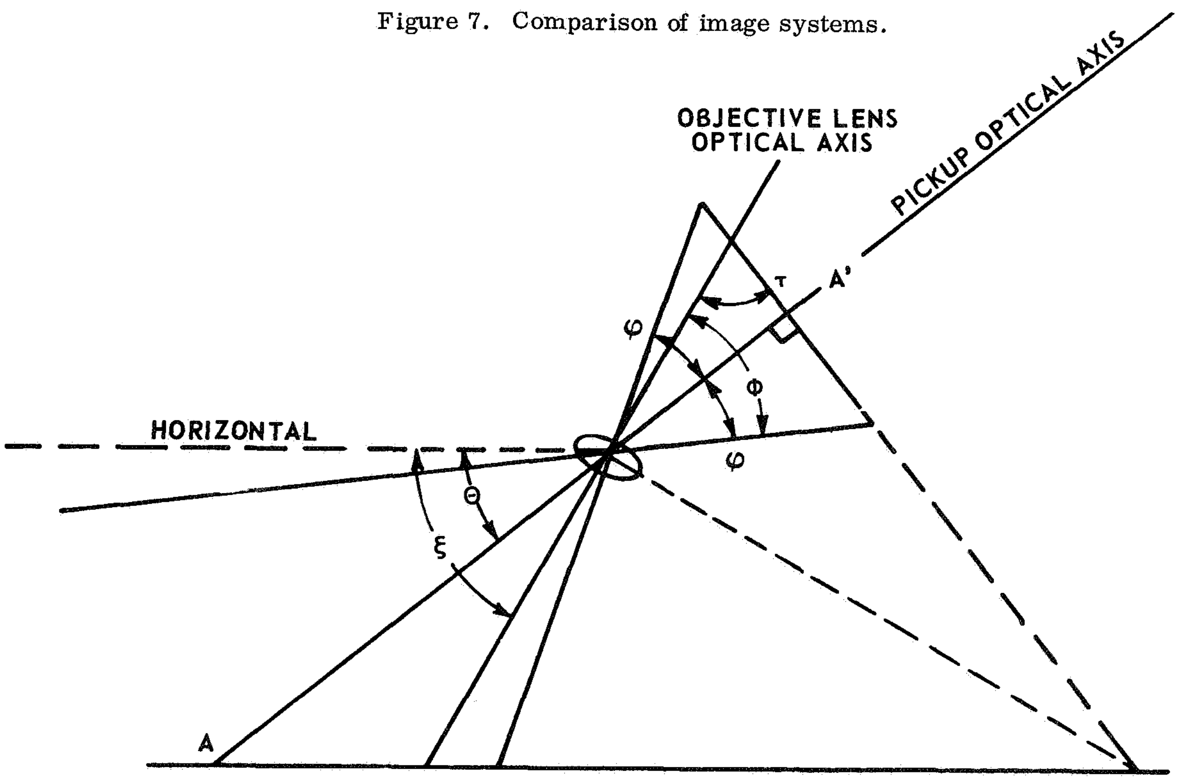


Figure 8. Tilted-objective lens technique.

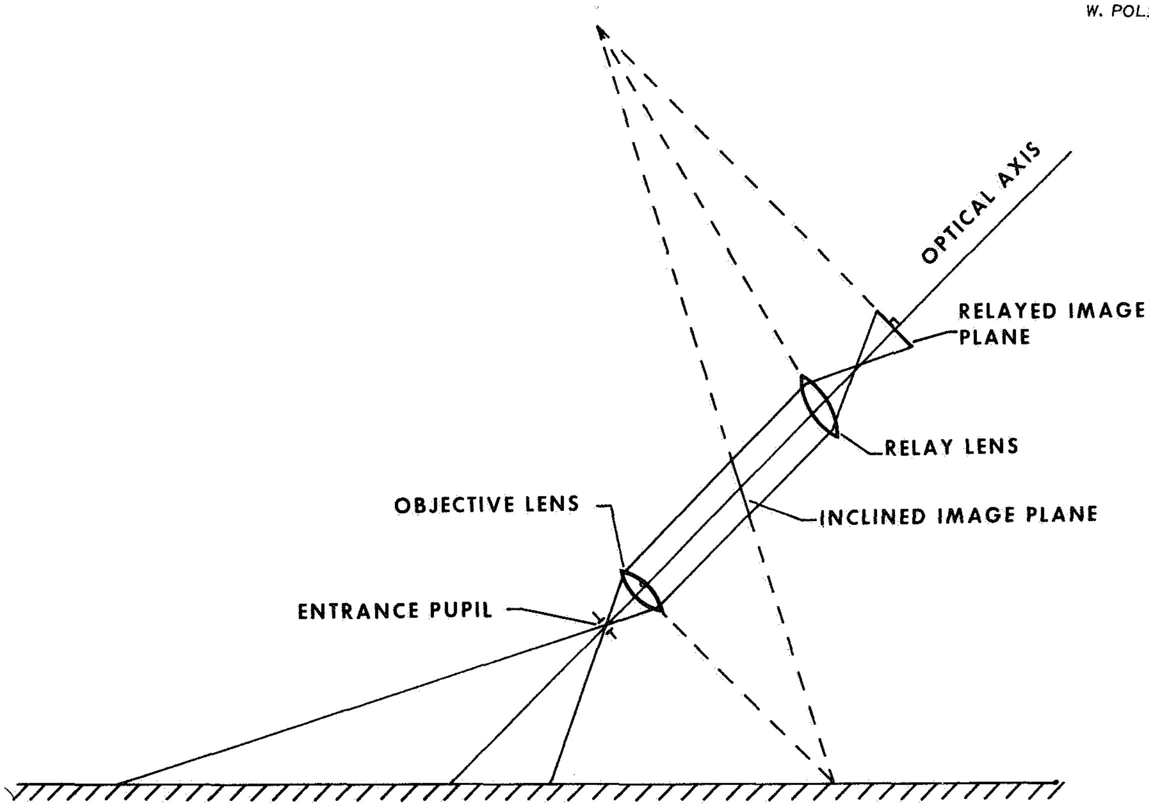


Figure 9. Relay lens correction technique.

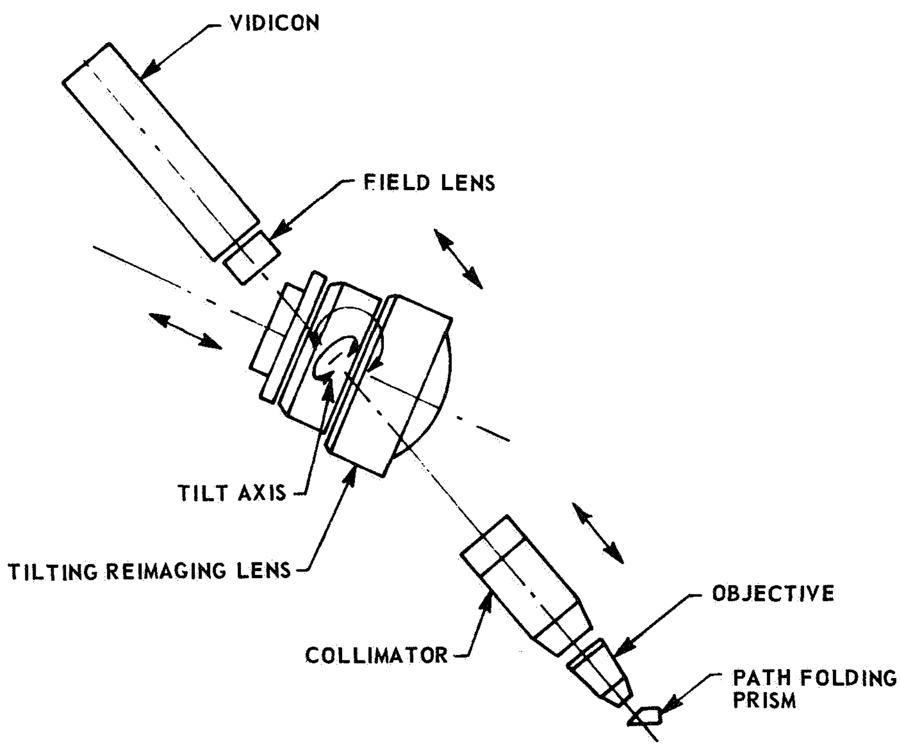


Figure 10. Goodyear probe, field-of-depth implementation.

COMPUTER-GENERATED VIEW OF A LUNAR LANDSCAPE

By

L. Thomas

Since the Simulation Branch of the Computation Laboratory of Marshall Space Flight Center first became involved in man-machine type simulators in 1962, much effort has been expended to provide better out-of-the-window views for the operator (or driver). In the past, visual cues have been provided using closed-circuit television. The television camera is normally mounted on a movable gantry and views a relief model of the vehicle environment. The generation of images in this manner requires:

1. Very large relief models for small areas.

2. Different models for each area simulated.

3. Position servos to provide system resolution.

4. Complex lighting arrangements.

5. High resolution closed-circuit television.

6. A terrain sensor for ground vehicles.

This system is not very versatile, and the out-of-the-window view provided is normally limited in resolution, field-of-depth, and field-of-view. Research has been done and is being done to decrease these limitations. In addition, ways have been sought that would eliminate the model completely. A research task was initiated to explore this possibility. The research was to be accomplished in two phases. The first phase was to generate television images by computers to complement present images. Some work was done in this area; that is, methods were investigated to visually display vehicle tracks and vehicle shadow. Also image enhancement techniques were investigated, but none were found to operate in real time except those already known in television technology.

The major emphasis was placed on phase 2, which was to devise methods of generating entire images by computers. Previous work had been done in this area, but this work always used composites of simple geometric shapes that were described by vector or elementary functions. No attempts had been made to generate real-time images with gray scale and texture.

It is not hard to visualize what a large task it would be to generate a real-time television image on a digital computer that would have the resolution required for simulation. For a minimum resolution of six lines or degrees over a 50- by 30-degree field at 30 frames/sec and with eight shades of gray would require 13 bits/μsec (Fig. 1). This is very high speed — too high for most computers. Therefore, methods were sought to compress data; i.e., to reduce the handling requirements of the digital computer. Hybrid computation was considered because it would permit the use of analog devices that are fast and can operate in parallel.

A scan converter is one example of such a device (Fig. 2). The converter accepts signals from the computer directly and allows them to be displayed in any format. The data to be displayed are transformed through coordinates reflecting the

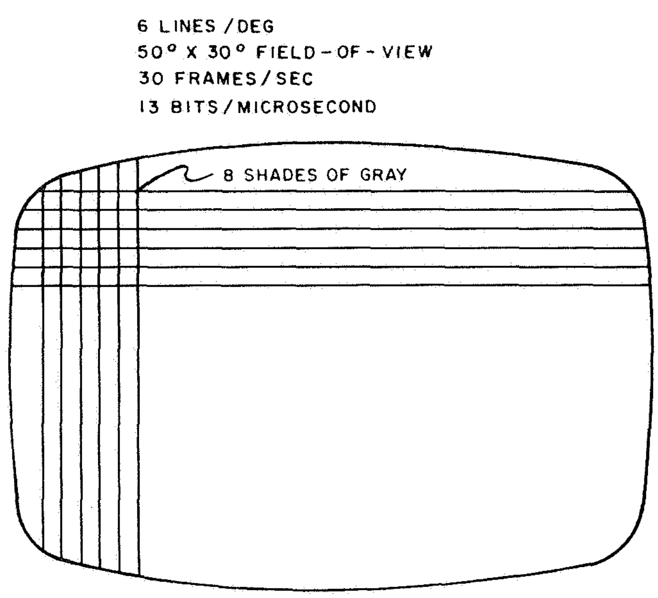


Figure 1. Digitized TV raster.

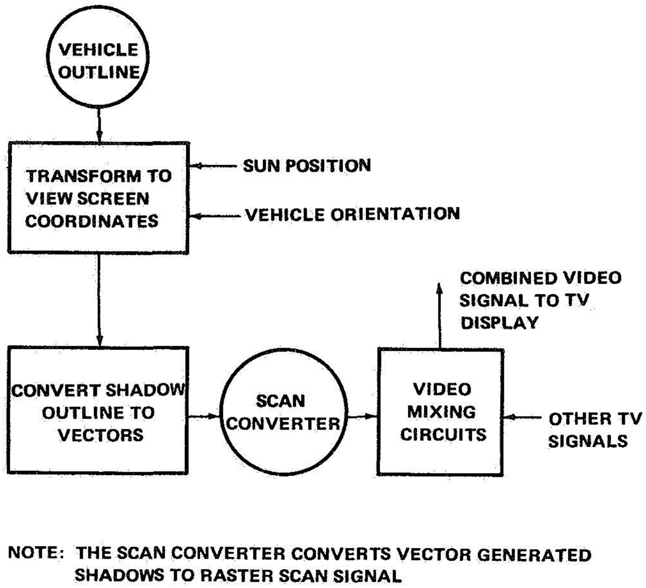


Figure 2. Use of scan converter.

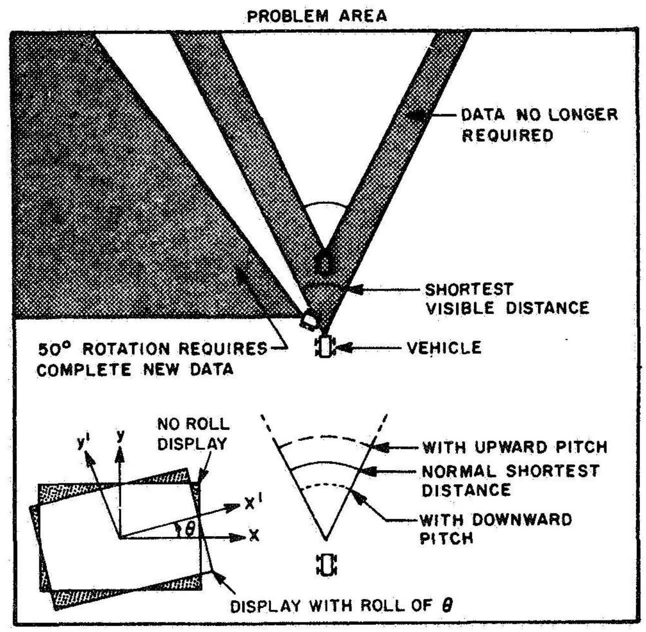


Figure 3. Change in data with vehicle motion.

sun position and vehicle orientation and are converted to vector form for the scan converter. The data are then read from the scan converter, possibly at a different rate and possibly in a different raster format, such as the standard for television monitors. It is also possible to erase and update any region of the scan without disturbing the rest of the picture. Therefore, by using a scan converter, advantage can be taken of compression techniques that actually exist in the real world.

Most of our simulator work with a man-in-the-loop has been studies involving lunar vehicles. Therefore, this was considered in the effort to reduce data handling. Some of the ways found to reduce data handling are (Fig. 3):

1. Forward motion requires rapid update of the foreground only.

2. The horizon changes rapidly in turns but requires less resolution.

3. A small amount of overscan can be carried to permit roll and pitch motions.

The repeatable characteristic of the moon lends itself readily to other techniques such as the following that can be used to further reduce computational requirements.

1. Height and slope data can be stored, not for each point on the terrain but by using two-dimensional polynomials. The surface to be represented can be divided into regions within which variations in height are neither too large nor too numerous. If actual data are used, a least-squares fit of a polynomial can be used. The coefficients of the polynomial then can be stored. This will allow different size regions to be used.

2. Craters can be classified into three general types; a simple rimless crater, a rim crater with a continuous curving bottom, and a rim crater with a flat bottom. The equations for these craters are available (Fig. 4). (Equations are only shown for crater (a) in Figure 4.)

3. Rocks and man-made features can be represented using polyhedrons. This requires that the vertices of the polyhedron and the unit normal to each face be stored (Fig. 5).

4. Texture can be produced using a random noise generator.

Theoretically, the visual display, or scene, is a projection of the light reflectivity of the terrain and must obey Lambert's law; that is, the light intensity from a surface element is a function of the angles to the sun and to the observer's eye (Fig. 6). To produce the display, each display element is projected down to the landscape and the light reflectivity from that area is used for the intensity

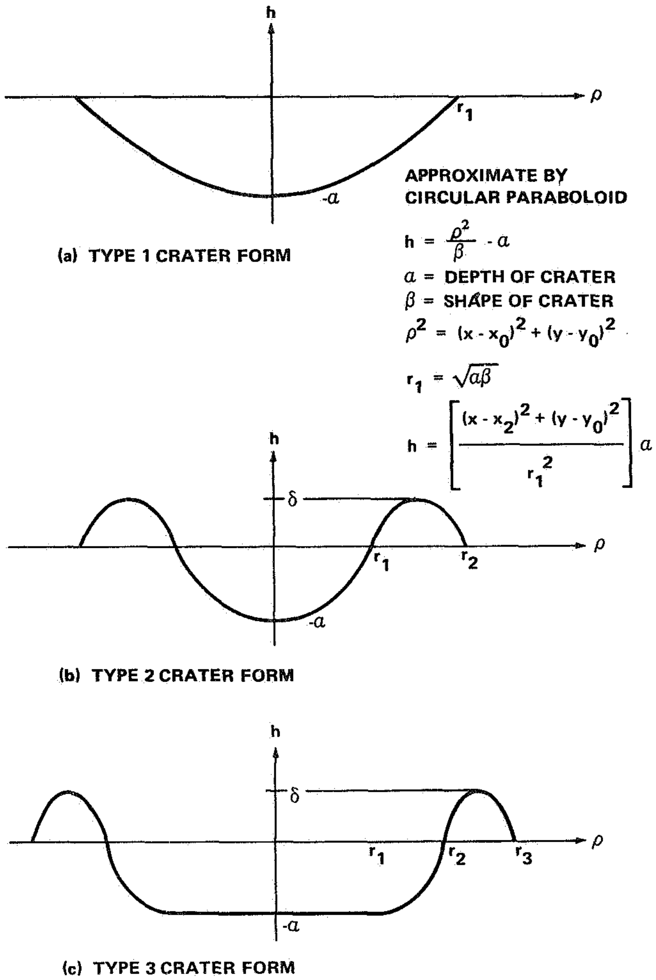


Figure 4. Crater functions.

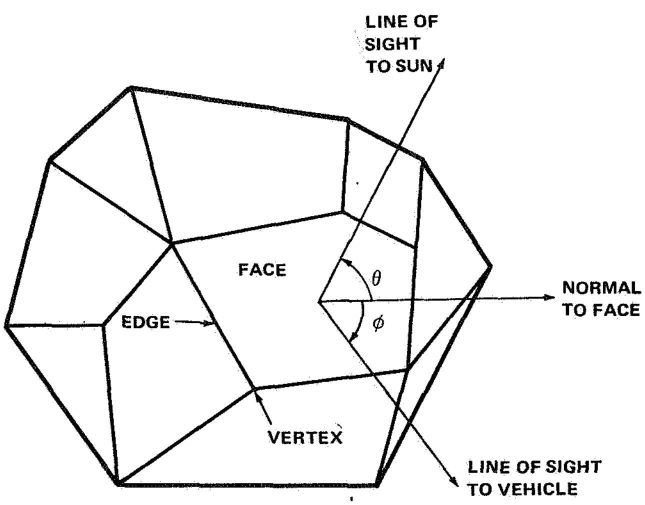


Figure 5. Illustration of a rock as a polyhedron.

of the display element. This projection procedure is complicated, since it must account for changes in vehicle orientation and for the varying height of the landscape. These varying heights cause part of the distant landscape to be blocked from view by near parts.

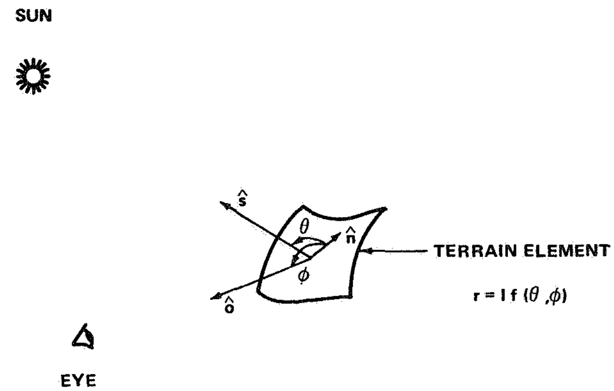


Figure 6. Calculation of specular reflection.

Remote shadows are computed using the relation shown in Figure 7. Areas that are blocked from view may be considered in the same manner if the reader considers the eye to be in the position in which the sun is shown on the figure. Remote shadows that depend on the terrain and incident angle of the sun can be computed and stored in memory before each run. So, in general, all the data needed for operation can be obtained. The position of the vehicle and its attitude can be computed continuously from the beginning of the run. Remote height and local slope data are readily obtainable from the polynomials. The sun position would normally be a fixed quantity for any single run, and as mentioned earlier, the remote shadow can be computed before each run.

Then, all of the elements mentioned can be combined as shown in Figure 8. The digital processing unit will output the quantities shown to the hybrid generators. Each element is then synchronized to one master, and each element is assigned certain priority. In other words, an image is never displayed if the computation shows it to be hidden from view.

In general, the system works as follows. At the beginning of a simulated run, the positions of the vehicle and the sun are entered into the computer. The remote shadows are then calculated before the run. After the run starts, the vehicle attitude and region of visibility are calculated and transferred to the hybrid generators which produce the height and reflectivity signals for the landscape along the projection of a display element, which is along the driver's line-of-sight. These signals are combined

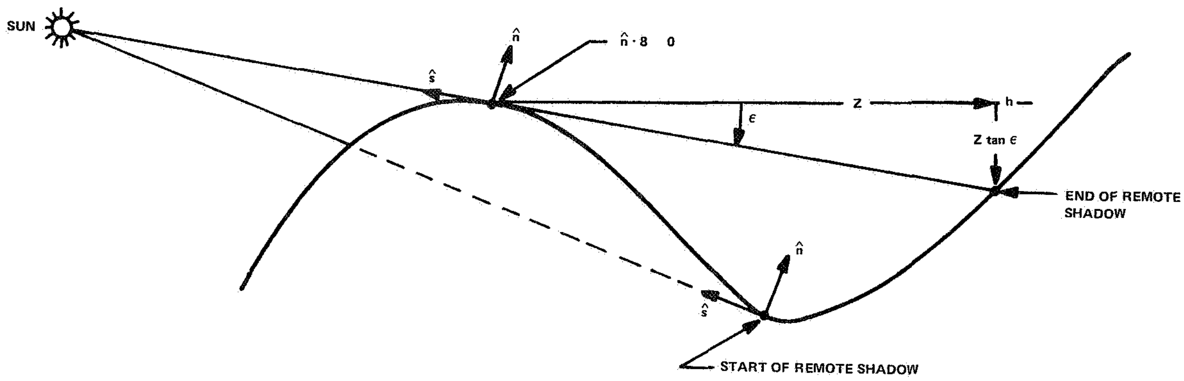


Figure 7. Graphic illustration of remote shadow calculation.

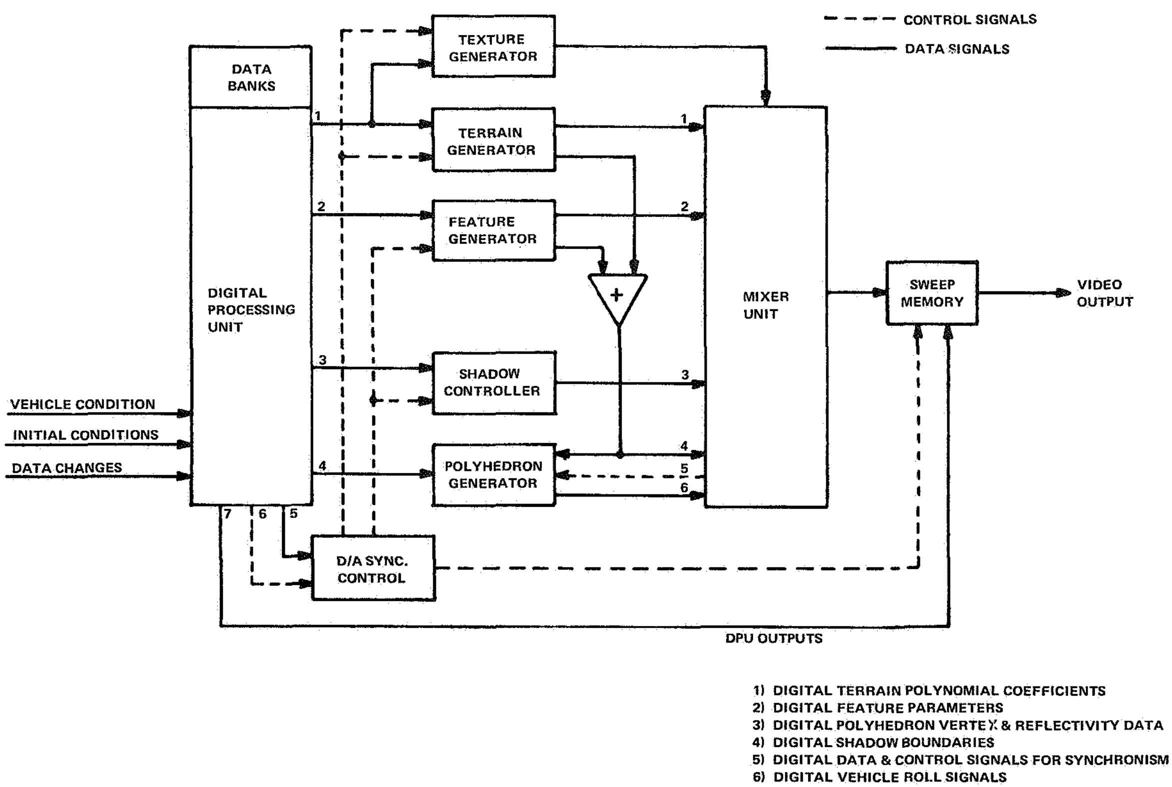


Figure 8. Overall system configuration.

in the mixer unit to produce a composite signal. This gives one set of intensities for a television raster sweep. The process is repeated for each sweep until the entire frame is completed. Then the data sent to the hybrid generators are revised, and the complete process is repeated.

This research program was intended to be in two parts, a feasibility study and an implementation study involving hardware and software. The second part of the program was never funded; therefore, this is not presented as a workable system. All questions have not been answered, but the system does offer merit for certain applications.

HYBRID SIMULATION OF VEHICLE RESPONSE TO MANY MEASURED WINDS

By

G. Prince

The program for Hybrid Simulation of Vehicle Response to Many Measured Winds is a computational support program performed by the Computation Laboratory for the Aero-Astrodynamics Laboratory of Marshall Space Flight Center. The Aero-Astrodynamics Laboratory has initiated several large analog simulations of Saturn boosters and is sponsoring a continuing effort in the area of statistical analysis of vehicle response to very large sets of actual wind data.

There is a large analog computer capability available at Marshall that has not been used to its full extent, primarily because of the difficulties of entering data and digesting the large quantities of information that the analog computer can produce. Most of the hybrid simulations performed up to this time have had a typical division of labor; i.e., high precision guidance and orbital equations on the digital computer, and control and high frequency dynamics on the analog computer.

Previously the simulation discussed here has been performed on a special-purpose, high-speed, repetitive-operation analog computer, since digital costs are prohibitive. However, because of parallel equipment limitations, it was necessary to simplify the model and run the same winds again and again and to compile the statistics largely by hand. This process requires several weeks to complete. It is now necessary to expand the model to a full six-degree-of-freedom, nonrigid body with highly nonlinear aerodynamics, complete automation of the statistical analysis, and a short turnaround time for a set of 2000 measured winds, so that parameter studies may be performed. This is an ideal problem for a modern hybrid system. The basic program has been completed and checked out. Although there have been many practical difficulties to overcome, valuable experience has been gained.

In the hybrid system process, the digital computer reads in from tape and pre-processes a single wind, and the analog computer solves the approximately 50 simultaneous differential equations representing the vehicle translations, rotations, engines, bending modes, etc., while calling on the digital computer for wind data and bivarient function generation only. At the end of each wind, the digital computer collects the statistical data that the analog has stored during the run and reads in another wind. After 2000 winds, the digital computer compiles the collected data and prints out the required statistics in the form of means, variances, exceedence counts, and probabilities.

A set of 2000 winds requires a running time of approximately 3 hours. Although it is capable of much higher speeds, the analog computer runs at 20 times real time because the digital computer holds it back. Every function possible has been assigned to the analog computer so that during a run, the digital computer is essentially performing bivarient aerodynamic function generation and automatically-loaded wind function generation. The speed of these operations can be increased by a factor of four by using machine-language programming, but it is necessary to use Fortran as long as there is a possibility of a major change in the mathematical model.

The following rough comparisons are made to give an idea of the costs and turnaround times involved in a simulation of this sort for 2000 winds.

	Analog	Digital	Hybrid
Time	2 weeks	50 hours	3 hours
Cost	$2500	$25 000	$250

The hybrid costs shown do not include hidden costs such as large amounts of time spent in programming, setup, and checkout; however, there are constant efforts within the industry to reduce time expenditures for these items. Considerable effort is also being exerted to develop special purpose digital hardware for analog function generation, which is the only time-critical job performed by the digital equipment in the current hybrid simulation.

Requests have already been received for hybrid simulation on new projects. If capabilities continue to develop in the area of hybrid simulation, many jobs now considered impossible for reasons of time or cost will become quite practical.

TELEMETRY LANGUAGE SYSTEM

By

R. Conway

A Telemetry Language System is being developed for the Instrumentation Checkout Complex (ICC) of the Quality and Reliability Assurance Laboratory of Marshall Space Flight Center. The primary requirement of the Telemetry Language System is to provide support for the ICC in all assigned missions. This system will provide a language to be used by both telemetry engineers and programmers in writing telemetry checkout procedures. Although the language is the key feature, the software system will provide a complete range of services including executive, loader, language processing, and all utility and support routines.

The Telemetry Language System will be capable of acquiring and recording any signal that can be received and routed by the Automatic Telemetry Ground Station (ATMGS). Evaluation of the telemetry data will be limited initially to post-test data.

The essential components of the Telemetry Language System will be the executive system, the telemetry language processor, and the support modules. Each of the system components will be interfaced such that a logical, smooth flow of control is attained. Each system component will realize the central control exerted by the executive. Each component must insure that control is relinquished to the executive upon termination and that system integrity is maintained. The executive will provide for all valid requests from called functions and will insure that the required services are provided to system devices and components. Functions called by the executive and linked by the executive will contain compatible interfacing or will interface indirectly through the executive.

Although the telemetry language will be designed specifically for ICC hardware and applications, the language processor will feature a high degree of generality and flexibility. The language processor could be adapted to provide support for other installations having similar hardware and applications. The majority of the telemetry language processor routines will be written in SDS META-SYMBOL for the SDS 930 computer and, therefore, will not be totally machine independent.

The telemetry language processor will process the source statements and directives and produce an executable program that will be capable of performing the required test procedure objectives.

The code generated by the telemetry language processor will consist of either inline code or calls and call sequences to external, closed subroutines. These subroutines will provide general support but will be designed for a specific application or for a particular system component. There will be a routine to provide support for each hardware system or subsystem. As an example, each subsystem of the ATMGS will be provided a support module that will be called when a program requires an interface with that particular hardware component. These support modules will be maintained on the system library magnetic tape and will be loaded and linked by the system loader.

Operationally, the Telemetry Language System may be divided into three components; hardware, software, and people (Fig. 1). The hardware component consists of the following:

1. SDS 930 computer
    a. Computer peripherals
        (1) Line printer
        (2) Two magnetic tapes
        (3) Paper tape reader
        (4) Paper tape punch
        (5) Card reader
        (6) Typewriter
    b. Two EJ-30 junction boxes
        (1) Multiplexer and analog-to-digital converter

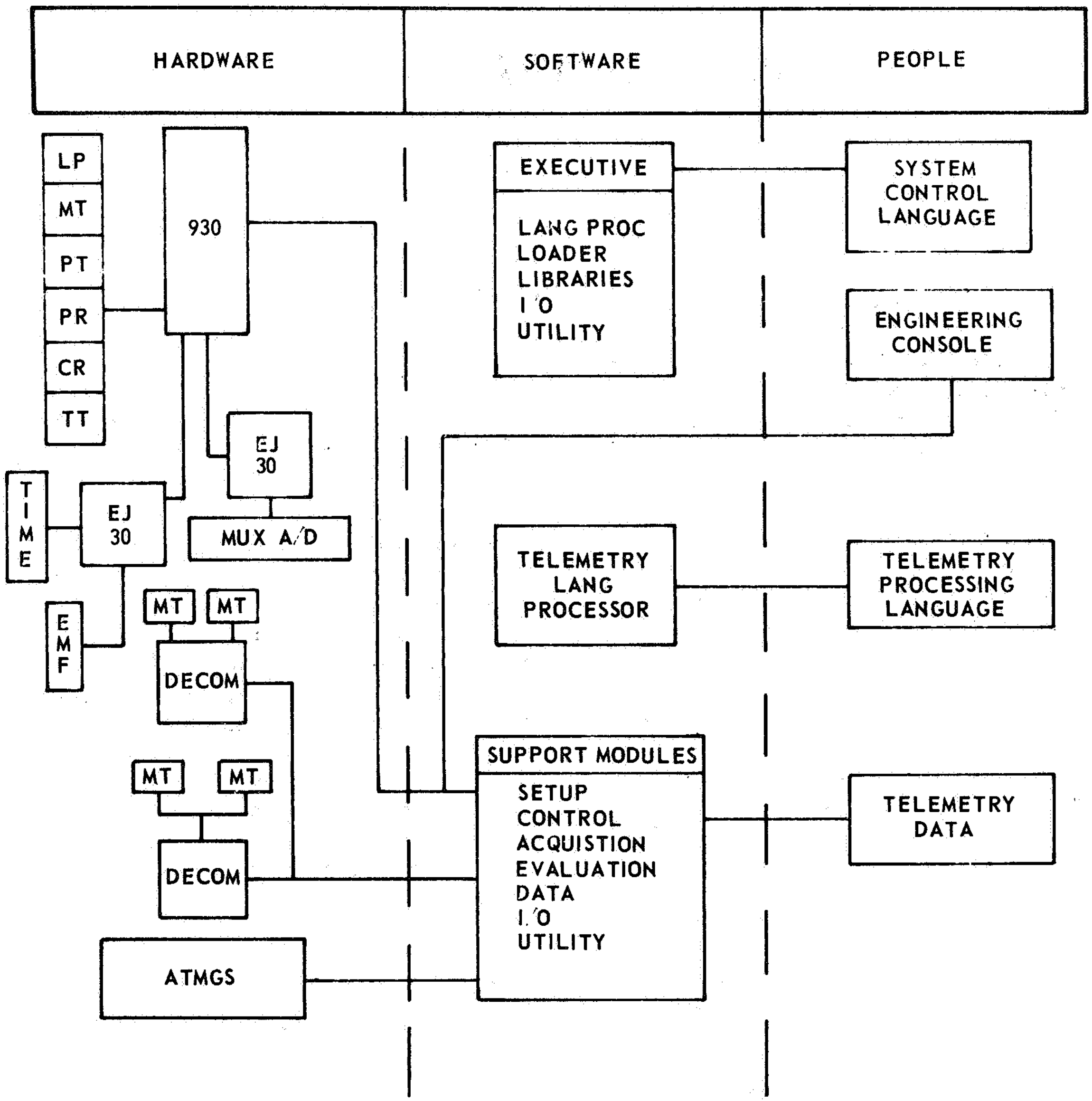


Figure 1. Telemetry language processor system.

(2) Real-time clock

(3) Interface with electromagnetic compatibility facility

2. Two decommutating buffers

a. Two magnetic tape drives

b. 24 digital-to-analog converters

3. Automatic Telemetry Ground Station

a. Receiver subsystem

b. Switching subsystem

c. Pulse code modulation subsystem

d. Discriminator subsystem

e. Oscillograph subsystem

The software component consists of the executive, the telemetry language processor, and the support modules. The executive provides the following:

1. Language processing functions (FORTRAN, ALGOL, META-SYMBOL, telemetry language)

2. Loading functions

3. Libraries (FORTRAN, arithmetic, and utility)

4. Input/output functions

5. Utility functions

One of the language processors will be the telemetry language processor. The support modules may be categorized by the function they perform such as:

1. ATMGS setup

2. DECOM control

3. Acquisition

4. Evaluation (control)

5. Telemetry data file maintenance

6. Input/output

7. Utility

The people component consists of

1. Operations personnel

2. Maintenance personnel

3. Telemetry engineers

4. Programmers

People communicate with the system with either system control language, telemetry processing language, telemetry data input, or the engineering console. Software communicates with the system through the executive, telemetry language processor, or support modules. The support modules communicate directly with the hardware in that there is a module to support each hardware component.

Of course, the key system interface for the telemetry engineer is the telemetry language processor. This component recognizes the command words and parameters of the telemetry language and compiles a telemetry test procedure from these.

Command words consisting of mnemonics and parameters for the telemetry language have been defined by the following categories. Each category may consist of several mnemonic operators, as shown in parentheses after each category.

1. ATMGS setup (12)

2. Decommutating buffer control (2)

3. Acquisition (1)

4. Control (8)

5. Arithmetic/logical (7)

6. Input/output (6)

7. MACRO (1)

8. Utility (6)

These 43 operators will provide capability for the telemetry engineer or programmer to configure the Instrumentation Checkout Complex, acquire telemetry data, evaluate that data, and record or display the results of that evaluation.

Figure 2 shows a sample program that was written in 30 minutes using the proposed telemetry language. This program will sample telemetry data and print the results after the sampled data have been compared to predicted data. These same functions written in assembly language could require from 9 to 12 man-months. Sixteen telemetry language statements were required for this example of a telemetry data scan. Three to four thousand statements would be required to perform the same function if this example was written in machine language.

However, a reduction of required effort would be attained only after the Telemetry Language System has been implemented. Total effort for design, development, and implementation of the Telemetry Language System has been estimated to require 3 to 4 man-years. For numerous requirements and applications, the savings in man efforts would be appreciable if support for the ICC was provided by this proposed system.

In summary it can be said that implementation of a telemetry language offers the most convenient method for establishing a high level interface between the telemetry engineer and the Instrumentation Checkout Complex. Using the language as a tool, an engineer is capable of directing the hardware into any selected configuration and is capable of performing any test function that is within the scope of the complex. When the language has been implemented and when it and all other system programs have been placed under the Telemetry Language System, a point of commonality will have been reached between the engineer and the programmer.

```
              SET       2
              TABLE     DATA (100), PRED (100)
              COUNT     C1 (1)

              READ      60, PRED, 100
              START     CONSOLE, 200
   200        SCAN      PRED, 100, 16, DATA
   201        COMP      DATA (C1), PRED (C1)
              JUMP      202
              WRITE     5, DATA (C1), 101
              WRITE     5, PRED (C1), 102
   202        INC       C1, +1, 100
              FIN

   100        FORM      1, 12, P, 100
   101        FORM      1, 6, V, 1
   102        FORM      0, 18, V, 1

              END
```

Figure 2. Sample program prepared using the proposed telemetry language.

SOME EXPERIMENTAL RESULTS CONCERNING THE ERROR PROPAGATION IN RUNGE-KUTTA TYPE FORMULAS

By

Erwin Fehlberg

This presentation has been prepared as a formal NASA Technical Report, TR R-352. Because of the complexity of the material, only the abstract has been reproduced here; however, the report is available at a cost of \$3.00 from the Clearinghouse for Federal Scientific and Technical Information, Springfield, Virginia 22151.

ABSTRACT

This report deals with the global error propagation of Runge-Kutta formulas. The problem is approached in two different ways. The first part presents the more conventional approach using the integrated differential equations for the error propagation. In the second part, two-sided (or bilateral) Runge-Kutta formulas are derived. Knowledge of the leading term of the local truncation error is essential for both approaches.

MARSHALL VEHICLE ENGINEERING SIMULATION SYSTEM (MARVES)

By

R. N. Setter

INTRODUCTION

The Marshall Vehicle Engineering Simulation System (MARVES) was first introduced in 1966 to standardize and simplify Marshall Space Flight Center's trajectory simulation programs. The present system consists of a problem-oriented programming language, a processor program that translates MARVES language statements into FORTRAN statements, and a subroutine library composed of trajectory computation subroutines and vector-matrix utility subroutines.

Although the MARVES language, with its associated processor, was designed for trajectory simulation, it is an excellent language for solving any set of ordinary differential equations by numerical methods. This general capability for solving differential equations has greatly increased the application of MARVES. A recently completed "machine independent" processor makes possible the installation of MARVES on any medium or large size digital computer.

The trajectory subroutine library was originally developed primarily for Saturn trajectory simulation; it is now being extended to meet Space Shuttle and Space Station requirements. The library is documented in the MARVES Trajectory Subroutine Library Manual. Volume I of this manual includes the subroutines designed for booster trajectory simulation. Volume II documents the routines for orbital and interplanetary flight simulation.

A conversational programming language system called SLAMS (Specification Language for Mission Studies) is now being developed as an extension to MARVES. SLAMS is designed to use remote site teletype or graphics terminals to specify the simulation mathematical models. The system then constructs the computer program required to solve the problem specified during the conversation.

MARVES LANGUAGE

The MARVES language is designed for solving differential equations with emphasis on the differential equations of trajectory simulation.

The solution of differential equations on a digital computer requires a numerical method of integrating the equations and a method of interrupting the integration to introduce discrete changes in the mathematical model. Initialization, termination, and auxiliary computations are also required. These requirements have led to the definition of the following five basic processes:

1. The initialization process consists of reading input data and computing starting conditions for the integration process and certain parameters that remain constant thereafter.

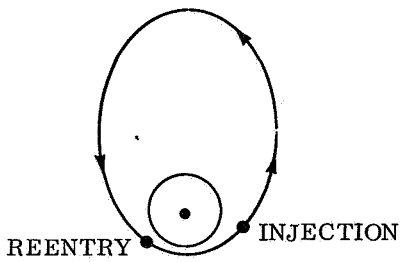
2. The evaluation of the differential equations.

3. The numerical integration process consists of providing a numerical procedure whereby the differential equations may be evaluated stepwise until stopping conditions have been reached.

4. The interrupt process consists of providing a method of interrupting the integration procedure when certain conditions are satisfied, or changes in dynamics are to be made.

5. The termination process consists of satisfying given stopping conditions and making needed terminal computations.

These processes form the basis for the MARVES language. The major statements of the language are INITIALIZE, DIFFERENTIAL EQUATIONS, INTEGRATION, and EVENTS (interrupt and termination). Figure 1 is a sample MARVES program of a simple trajectory problem. This example



INJECT INTO ATMOSPHERE-INTERSECTING ORBIT
COMPUTE TRAJECTORY TO REENTRY AND STOP
COMPUTATION AT MAXIMUM AERODYNAMIC PRESSURE

```
→        INITIALIZE
             INPUT NAMELIST (5, DATA) R, V, ---

→        DIFFERENTIAL EQUATIONS
     C   COMPUTE EARTH'S GRAVITATIONAL FORCE
             CALL GRAVT1
     C   COMPUTE AERODYNAMIC DRAG FORCE
             CALL PRA63
             CALL RELVEL
             CALL AEROF
     C   SUM FORCES
             CALL SUMF (2, A)

→        INTEGRATION
         METHOD RUNGE KUTTA ORDER = 2, R(V), V(V), A(V)

→        EVENTS
         EVENT (EXIT) Q = MAXIMUM
         OUTPUT LIST R(V), V(V)

         TERMINAL COMPUTATIONS
         STOP
         END
```

Figure 1. Sample MARVES program.

demonstrates the major language statements. Several numerical integration methods are available with the system. These methods include Runge Kutta, Fehlberg's, Shank's, and Butcher's and are specified in the language, by name, following the integration METHOD statement. The MARVES language has much more capability than is shown in this simple example. The details of the language are documented in the MARVES User's Manual.

MARVES PROCESSOR PROGRAM

The function of the MARVES processor program is to convert MARVES source language into FORTRAN code. This FORTRAN coding is then processed and executed by the computer operating system in the same manner as any other FORTRAN source coding.

The MARVES processor is written in ASA Standard FORTRAN Code with special attention given to making the program machine independent. This machine independence is demonstrated at MSFC by the fact that MARVES is operational on the Raytheon 520, the EMR 6050, the UNIVAC 1108, and the IBM 7094.

The processor is modular in design so that the full system can be installed on large computers and a limited version can be installed on smaller machines. As a result of the machine-independent, modular design of the MARVES processor, it can be installed on any machine with at least two external scratch files and an approximately 20K memory.

MARVES SUBROUTINE LIBRARY

The MARVES subroutine library is composed of trajectory simulation subroutines and vector-matrix utility subroutines. Until recently the library was somewhat specialized for Saturn vehicle simulation. It is presently being overhauled to meet requirements for such projects as the Space Shuttle and Space Station.

Recent subroutine library documentation separates the library into subroutines designed for booster trajectory simulation (Volume I) and subroutines for orbital and interplanetary flight (Volume II). Table 1 shows the subroutine categories as documented in Volume I. All the vehicle-dependent subroutines are grouped into one category. The majority of the subroutines of the library are vehicle-independent and are designed to be used in simulating future vehicles such as the Space Shuttle. Volume II of the Trajectory Subroutine Library Manual documents all the subroutines required to simulate orbital and interplanetary trajectories. It also documents the routines that relate the trajectories to tracking stations, data acquisition stations, ground targets, and celestial targets. These subroutines are presently being used for Space Station studies.

TABLE 1. MARVES TRAJECTORY SUBROUTINE LIBRARY

- Environment Simulation Subroutines
- Vehicle Simulation Subroutines
- Trajectory Computation Subroutines
- Coordinate Transformation Subroutines
- General Utility Subroutines
- Vehicle Dependent Subroutines

SLAMS EXTENSION TO MARVES

SLAMS (Specification Language for Mission Studies) is being developed to introduce a conversational programming language system to simulate trajectories for mission studies. SLAMS is an extension of the existing MARVES system. It makes use of the MARVES preprocessor and subroutine library. The present SLAMS system includes the capability to specify a trajectory simulation in terms of environment model, vehicle model, and desired end conditions. A more complete trajectory design system allowing specification of simulations from mission requirements is now under study.

The problem area of trajectory simulation was chosen for implementing a conversational language system because of the large amount of trajectory simulation done at MSFC and because of the large variety of simulation options desired. Options for simulating Saturn V and Saturn IB vehicles are included; however, the system is designed to be extended to include simulation capability for any future vehicle.

The SLAMS system includes a special purpose dialog that is used to construct the desired program from a remote site teletype terminal. Special purpose language systems, such as SLAMS, are intended to extend conversational computing capability rather than replace the general purpose conversational programming languages. General purpose systems, such as conversational FORTRAN, include dialog that is mainly diagnostic (compilation error, etc.). More specialized systems, such as AMTRAN, have been developed to bring the programming language closer to the mathematics. These general purpose systems serve well for computational tasks that can be defined from basic mathematical tools.

Many computational problems facing scientists and engineers today are so large and complex that it is difficult to define their solution from basic mathematical routines while working at a computer console. For these large computational problems, the conversational capabilities of the computer system can be used more effectively in guiding the construction of the simulation programs. To accomplish this, the SLAMS system is designed so that the computer asks specific questions about the desired simulation until it has the necessary information to construct the required program. The computer then prints a list of the input parameters needed to run the program. Execution can begin immediately if the input is known, or at a later time if the data must be acquired.

The conversation is a method of selecting simulation options. Subroutines are available in the SLAMS library for each option in the system. The simulation can be specified exactly if it is within the capability of the system. It is, however, unrealistic to assume that options could be included to cover all desired simulations. For this reason, the system was designed for easy program alteration. The conversation can be used to get as close as possible to the desired simulation. The user can then, making use of the complete documentation of the subroutines used in the generated program, make alterations or add new subroutines with relative ease.

Some of the advantages of constructing programs with a conversational system are:

1. A program can be created by an individual familiar with the problem area that does not want to become involved with detailed programming.

2. Programming and debug time should be shorter than with conventional programming methods.

3. The conversational system generates a "clean" program that is "checked out" for a specific mathematical model and contains only program elements relevant to the specified model.

4. By using the conversational system to construct programs, a high degree of program standardization can be achieved. The system provides the capability to generate programs tailored to a specific problem that are made up of standard elements interchangeable with other programs generated by the system.

A summary of the general steps required to operate the present SLAMS system from an 1108 teletype and remote terminal follows:

1. From any 1108 system teletype, type in a RUN statement and execute SLAMS.

2. Select simulation options by answering questions presented by the SLAMS conversational interpreter. A final question will ask whether the user desires to catalog the generated MARVES program or punch the program on cards.

3. Tear off the printed copy of the dialog and the required input listing and save for reference.

4. Execute the generated program from the Data Communications Terminal (9200 or 9300). Before execution, the program may be modified to include special capability not included in the system.

APPROVAL

TM X-64527

RESEARCH ACHIEVEMENTS REVIEW
VOLUME III REPORT NO. 12

The information in these reports has been reviewed for security classification. Review of any information concerning Department of Defense or Atomic Energy Commission programs has been made by the MSFC Security Classification Officer. These reports, in their entirety, have been determined to be unclassified.

These reports have also been reviewed and approved for technical accuracy.

H. Hoelzer

DR. H. HOELZER
Director, Computation Laboratory

CALENDAR OF REVIEWS

FIRST SERIES (VOLUME I)

REVIEW	DATE	RESEARCH AREA	REVIEW	DATE	RESEARCH AREA
1	2/25/65	RADIATION PHYSICS	12	9/16/65	AERODYNAMICS
2	2/25/65	THERMOPHYSICS	13	9/30/65	INSTRUMENTATION
3	3/25/65	CRYOGENIC TECHNOLOGY	14	9/30/65	POWER SYSTEMS
			15	10/28/65	GUIDANCE CONCEPTS
4*	3/25/65	CHEMICAL PROPULSION	16	10/28/65	ASTRODYNAMICS
5	4/29/65	ELECTRONICS	17	1/27/66	ADVANCED TRACKING SYSTEMS
6	4/29/65	CONTROL SYSTEMS	18	1/27/66	COMMUNICATIONS SYSTEMS
7	5/27/65	MATERIALS			
8	5/27/65	MANUFACTURING	19	1/6/66	STRUCTURES
9	6/24/65	GROUND TESTING	20	1/6/66	MATHEMATICS AND COMPUTATION
10	6/24/65	QUALITY ASSURANCE AND CHECKOUT	21	2/24/66	ADVANCED PROPULSION
11	9/16/65	TERRESTRIAL AND SPACE ENVIRONMENT	22	2/24/66	LUNAR AND METEOROID PHYSICS

SECOND SERIES (VOLUME II)

REVIEW	DATE	RESEARCH AREA	REVIEW	DATE	RESEARCH AREA
1	3/31/66	RADIATION PHYSICS	7	3/30/67	CRYOGENIC TECHNOLOGY
2	3/31/66	THERMOPHYSICS	8**	5/25/67	COMPUTATION
3	5/26/66	ELECTRONICS	9	7/27/67	POWER SYSTEMS
4	7/28/66	MATERIALS	10	9/28/67	TERRESTRIAL AND SPACE ENVIRONMENT
5	9/29/66	QUALITY AND RELIABILITY ASSURANCE	11	11/30/67	MANUFACTURING
6	1/26/67	CHEMICAL PROPULSION	12	1/25/68	INSTRUMENTATION RESEARCH FOR GROUND TESTING

THIRD SERIES (VOLUME III)

REVIEW	DATE	RESEARCH AREA	REVIEW	DATE	RESEARCH AREA
1	3/28/68	AIRBORNE INSTRUMENTATION AND DATA TRANSMISSION	6	1/30/69	THERMOPHYSICS
			7	3/27/69	RADIATION PHYSICS
2	5/22/68	ASTRODYNAMICS, GUIDANCE AND OPTIMIZATION	8	6/26/69	METEOROID PHYSICS
			9	9/25/69	COMPUTATION RESEARCH
3	7/25/68	CONTROL SYSTEMS	10	12/18/69	MATERIALS RESEARCH FOR SHUTTLE AND SPACE STATION
4	9/26/68	AEROPHYSICS			
5	11/21/68	COMMUNICATION AND TRACKING	11	1/29/70	MICROELECTRONICS RESEARCH FOR SHUTTLE AND SPACE STATION
			12	3/26/70	COMPUTATION RESEARCH

* Classified. Proceedings not published.

** Proceedings summarized only.

Correspondence concerning the Research Achievements Review Series should be addressed to:
Research Planning Office, S&E-R, Marshall Space Flight Center, Alabama 35812